Working-memory failure in phone-based interaction

Full text Pdf (561 KB)

Source ACM Transactions on Computer-Human Interaction (TOCHI) archive
Volume 4 , Issue 2 (June 1997) table of contents
Pages: 67 - 102
Year of Publication: 1997
ISSN:1073-0516

Authors Brian R. Huguenard Univ. of Notre Dame, Notre Dame, IN
F. Javier Lerch Carnegie Mellon Univ., Pittsburgh, PA
Brian W. Junker Carnegie Mellon Univ., Pittsburgh, PA
Richard J. Patz Carnegie Mellon Univ., Pittsburgh, PA
Robert E. Kass Carnegie Mellon Univ., Pittsburgh, PA

Publisher ACM Press New York, NY, USA

Additional Information: abstract references citings index terms review collaborative colleagues peer to peer

Tools and Actions: Discussions Find similar Articles Review this Article
Save this Article to a Binder Display in BibTex Format

DOI Bookmark: Use this link to bookmark this Article: http://doi.acm.org/10.1145/254945.254947
What is a DOI?

ABSTRACT

This article investigates working-memory (WM) failure in phone-based interaction (PBI). We used a computational model of phone-based interaction (PBI USER) to generate predictions about the impact of three factors on WM failure:PBI features (i.e. menu structure), individual differences (i.e., WM capacity), and task characteristics (i.e., number of tasks). Our computational model stipulates that both the storage and the processing of information contribute to WM failure. In practical terms the model and the empirical results indicate that, contrary to guidelines for the design of phone-based interfaces, deep menu hierarchies (no more than three options per menu) do not reduce WM error rates in PBI. At a more theoretical level, the study shows that the use of a computational model in HCI research provides a systematic approach for explaining complex empirical results.

REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 AGRESTI, A. 1990. Categorical Data Analysis. Wiley, New York.

2 John R. Anderson, The Architecture of Cognition, Harvard University Press, Cambridge, MA, 1983

Working-Memory Failure in Phone-Based Interaction

BRIAN R. HUGUENARD
University of Notre Dame
and
F. JAVIER LERCH, BRIAN W. JUNKER, RICHARD J. PATZ, and ROBERT E. KASS
Carnegie Mellon University

This article investigates working-memory (WM) failure in phone-based interaction (PBI). We used a computational model of phone-based interaction (PBI USER) to generate predictions about the impact of three factors on WM failure: PBI features (i.e., menu structure), individual differences (i.e., WM capacity), and task characteristics (i.e., number of tasks). Our computational model stipulates that both the storage *and* the processing of information contribute to WM failure. In practical terms the model and the empirical results indicate that, contrary to guidelines for the design of phone-based interfaces, deep menu hierarchies (no more than three options per menu) do not reduce WM error rates in PBI. At a more theoretical level, the study shows that the use of a computational model in HCI research provides a systematic approach for explaining complex empirical results.

Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine—*human information processing;* H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*evaluation; theory and methods*

General Terms: Experimentation, Performance

Additional Key Words and Phrases: Auditory menu, cognitive model, individual differences, user error, working memory

The work of Lerch and Huguenard was supported in part by Bellcore grants 1-41117 and 1-41239. The work of Junker and Patz was supported in part by the Office of Naval Research, Cognitive Sciences Division, grant N00014-91-J1208, R&T 4421-560.

Authors' addresses: B. R. Huguenard, Department of Management, University of Notre Dame, Notre Dame, IN 46556-0399; email: brian.huguenard.1@nd.edu; F. J. Lerch, Graduate School of Industrial Administration and Human Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890; email: lerch@andrew.cmu.edu; B. W. Junker, R. J. Patz, and R. E. Kass, Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213; email: {junker; patz; kass}@stat.cmu.edu.

1. INTRODUCTION

Working-memory limitations have been recognized as a major bottleneck in human information processing since the onset of cognitive research [Broadbent 1958; Miller 1956; Peterson and Peterson 1959]. Working memory (WM) is defined by Baddeley [1986] as "... a **system** for the temporary **holding** and **manipulation** of information during the performance of a range of cognitive tasks such as comprehension, learning and reasoning" (boldface added). There is a general agreement that many human errors in problem solving and decision making can be traced to WM failure [Anderson 1990]. Recently there have been serious efforts to devise cognitive architectures that explain WM phenomena in terms of both the storage and the processing of information [Baddeley 1986; Just and Carpenter 1992; Schneider and Detweiler 1988].

This research is conducted by building a computational model (PBI USER) based on the CAPS cognitive architecture [Just and Carpenter 1992], using this model to generate predictions of WM failure in a simple domain (phone-based interaction) and testing these predictions with human subjects. Several studies have shown the impact of WM failure on error behavior in computer-based tasks such as computer programming [Anderson and Jeffries 1985; Kessler and Anderson 1986], spreadsheet construction [Lerch et al. 1989; Lerch 1994], and database query writing [Smelcer 1989; Carlin et al. 1992]. None of these studies modeled WM demands in terms of both storage and processing. PBI USER models the simultaneous demands of processing and storing information in WM and makes predictions of WM failure based on these demands.

A better understanding of the factors contributing to WM failure should have a substantial impact on current practices for interface design. Prior to the advent of computer technology, methods for executing formal problem-solving tasks usually did not generate high WM loads. For example, methods for executing multicolumn addition and subtraction with pencil and paper use external aids (e.g., writing the carry digits) to minimize WM load. Similarly, we learn to solve physics problems by building diagrams that lessen WM load. These methods were developed through trial and error over long periods of time. The abrupt arrival of computer technology has generated methods (i.e., software) that often generate very high WM load. Some examples are writing formulas with spreadsheet software using nonmnemonic cell references, programming in LISP, writing multitable queries in SQL, and of course, interacting with computers through the telephone. Consequently, through the design of user interfaces that do not generate high WM loads, software usability and user acceptance should be improved.

The organization of the article is as follows. Section 2 reviews prior research in WM and describes PBI as a high WM load task. PBI USER and the research hypotheses generated by running PBI USER are presented in Section 3. Sections 4, 5, and 6 describe the experiment with human subjects, the statistical analysis, and the experimental results. Section 7 discusses these results in relation to the predictions made by PBI USER. It also presents an interpretation of the results that failed to support our hypotheses and discusses these results in terms of modifications made to PBI USER. Finally, Section 8 presents implications for human-computer interaction.

2. WORKING MEMORY AND PHONE-BASED INTERACTION

The dominant models of working memory in the 60's were based on the underlying assumption that WM consisted of a small number of fixed slots in which information could be temporarily held. Miller's [1956] classic paper "The Magical Number Seven" stimulated interest in the limits of WM capacity, which he demonstrated to be centered around seven items (or chunks) of information. Waugh and Norman [1965] developed a model of human memory that included a short-term store (their version of WM) with a small number of fixed slots. In their model, items enter the short-term store and can get lost either by decay over time or by being displaced by new items. In order to counteract the decay process, items can be maintained in the short-term store by rehearsal. A similar model was developed by Atkinson and Shiffrin [1968] in which they differentiated between processing structure (the involuntary processes through which information is processed) and control processes (strategies for handling information under voluntary control). Although this last model distinguishes between storage and processing of information, Atkinson and Shiffrin's [1968] model portrayed WM mainly as a *storage* mechanism, while not fully addressing the role of WM in the *processing* of information.

Baddeley and Hitch [1974] began to modify the view of a fixed slot WM with a model that assumes a central executive and two slave processors: an articulatory loop and a visual-spatial sketch pad. While the articulatory loop stores basically verbal information, the visual-spatial sketch maintains and manipulates visual-spatial images. The central executive coordinates information from the two slave processors, allocates attention, and is the medium for control processes [Atkinson and Shiffrin 1968]. This model emphasizes the dual purpose of WM, that is, the storage and processing of information. Although the model has provided insights into an impressive number of experimental results, it has not been implemented in a computational form [Card 1988].

Just and Carpenter [1992] have proposed a theory of WM in which *both* storage and processing demands determine WM load and the probability of WM failure. This theory has been instantiated in a computational cognitive architecture called CAPS. CAPS is a hybrid of a production system and an activation-based connectionist system and is described by Just and Carpenter [1992] as "... a computational theory in which both storage and processing are fueled by the same commodity, namely activation. In this framework, capacity can be expressed as the maximum amount of activation available in working memory to support either of the two functions." PBI USER is based on this cognitive architecture of WM.

Activation is the general WM resource consumed in CAPS by the processing and the storage of information in WM. Each WM element in a CAPS model has an associated activation level which can be conceptualized as the current strength of the memory-trace of the element. In order for a WM element to be accessible for retrieval or manipulation, the activation level of the element must be above some minimum threshold value. Productions within a CAPS model can be used to alter the activation level of WM elements through a process called *directed activation,* in which the productions specify increments or decrements of activation to be applied to specific WM elements. This is to be contrasted with the process of *spreading activation* [Anderson 1983; Collins and Loftus 1972], in which activation is spread from a source node to all connected nodes (and then from those nodes to all connected nodes, etc.) in the declarative knowledge base.

Activation is used in CAPS (1) for maintaining WM elements so they are not forgotten and (2) for processing WM elements. A key feature of the architecture is that the total amount of activation available can be constrained to a preset amount called WM capacity. Given this constraint, if the total processing and storage demands for activation exceed the total amount of WM capacity, then all activation is scaled back so that the activation constraint is not exceeded. This "scaling back" has the effect of slowing down processing (since less activation is spread per unit of time, more time will now be required to complete the processing) and the effect of causing a form of forgetting through displacement (since less activation is now available for storage, some WM elements may lose so much activation that they are effectively forgotten). It is important to note that it is the combined demand of storage and processing for a given task that determines whether or not the activation constraint (WM capacity) will be exceeded. Therefore this combined demand determines the probability of WM failure.

Several studies have shown that CAPS is capable of explaining and predicting human performance in a variety of complex tasks with substantial WM demands. CAPS was initially used for the simulation of word-by-word gaze duration in text comprehension by human readers [Thibadeau et al. 1982]. CAPS was also used to simulate response latencies, eye-fixation patterns, and retrospective strategy reports in the performance of psychometric mental-rotation tasks [Just and Carpenter 1985]. Finally, Just and Carpenter [1992] modified the original CAPS model to account for individual differences in text comprehension performance due to differences in WM capacity. In summary, CAPS has demonstrated that it is capable of predicting individual performance differences in tasks that involve processes similar to those encountered in phone-based interaction (e.g., language comprehension and perceptual processes).

The task chosen for the current study is phone-based interaction. Phone-based interfaces operate by allowing the user to enter commands over the telephone keypad or by voice recognition and by providing output in the form of recorded or synthesized speech. These interfaces can be categorized as *command based* or as *prompted.* The *command-based* interface does not provide prompts that would indicate the choice of commands available to the user, but instead assumes that the user knows the commands. The *prompted* interface typically involves a hierarchical structure of voice menus, in which a voice prompt presents a menu of different actions that can be taken, along with the corresponding key that should be pressed (or words that should be spoken) in order to choose each action. By choosing the appropriate option at each menu presentation (called a *choice point*), the user can navigate through the hierarchy of menus until the desired final action has been performed. Because the user is reminded of the available options at every choice point, prompted interfaces are appropriate for applications that are used infrequently and for applications for which users receive minimal training (interfaces for such applications are often called *walk-up-and-use* interfaces).

The different styles of PBI (*command based* versus *prompted, keypad based* versus *voice recognition,* etc.) would create different processing and storage loads on WM. This study focuses on *prompted, keypad-based* PBI because this is a very common interface. It is also a high WM load task, and it is a relatively simple task to model (for the experimental materials and for the cognitive model). In storage and processing terms, interactions encountered during this type of PBI generate WM overload in at least three different ways.

First, the user must monitor the status of a constantly changing environment (i.e., the presentation of options for each menu). Changes in the environment must be noted by the user because they allow the evaluation of the alternatives for the next option choice and because they provide feedback on the efficacy of prior option choices. Second, pacing of information presentation (i.e., presentation of menu options) is done by the interface. This pacing generates time pressures that increase WM processing demands. Third, external memory aids (which help to enhance effective WM capacity) are usually not present; consequently, the user is required to keep track in WM of the degree of "goodness" of alternative options that have already been considered while processing the information about the current option.

PBI has considerable advantages as a task domain for building detailed cognitive models of WM failure and for empirically investigating error behavior due to WM limitations. First, PBI is conceptually simple, so knowledge of the task is easy to acquire for the subjects and simple to represent in the cognitive model. This allows us to concentrate our effort on how information is encoded, maintained, and processed in WM by the user. Second, PBI allows direct experimental manipulation of WM load. For example, PBI users can be given tasks with or without modifiers (e.g., "retrieve schedule" versus "add class 247," where 247 is a modifier indicating the identification number of an academic course), or users can be asked to retain information about two tasks and to perform them consecutively. Third, PBI allows only two types of execution errors: (a) *information loss* *errors* in which users forget information about the task[1] (e.g., the code of the class to be added, the name of the object for which information is to be retrieved) and (b) *choice errors* in which users select the wrong alternative when presented with a set of choice options. Although data entry errors are a third possible type of execution error, we assume that use of a touch-tone keypad is a well-learned skill and that few such errors occur. (In our experiment the buttons of the simulated telephone keypad are oversized, and the interbutton gaps are large, while the mouse-controlled pointer used to select buttons is relatively small. This makes it impossible to accidently press two keys simultaneously, as can happen with fingers and standard-sized keypads.)

In terms of the CAPS architecture, the first type of PBI execution error—information loss—involves the *storage* of information in WM and is directly caused by WM capacity being insufficient to satisfy the activation demands of storing essential task-specific information. The second type of execution error, choice error, involves the *processing* required to select an alternative from a set of choices. A choice error can be caused by (1) WM capacity being insufficient to satisfy the activation demands of the processing needed to make the correct choice with the available information and/or (2) lack of knowledge about how the goal (the task to be accomplished) relates to the current set of choice alternatives.

During the experiment, subjects were given a PBI task to perform and were required to store the task description in WM during task execution. In terms of the experimental procedure, information loss errors (involving a failure of WM storage) were indicated by the subject requesting an additional presentation of the task description (after task execution had begun). Choice errors (involving a failure of WM processing and/or a lack of knowledge) were indicated by the subject making one or more incorrect option choices while traversing the menu hierarchy. More details concerning the experimental procedure are given in Section 4.4.

3. PBI USER AND RESEARCH HYPOTHESES

We have developed a computational model (PBI USER) based on the CAPS architecture that simulates the storage and processing of information in WM in phone-based interaction. As shown in Figure 1, PBI USER has two main components: the Device Model and the User Model. The Device Model, written in Lisp, simulates the behavior of a specific telephone interface. It contains (1) the topology of the menu structure of the phone-based interface and (2) the phrases that specify the options at each level of the menu structure. The User Model is a production system written in CAPS and simulates the WM processes of a user interacting with the Device Model.

[1] Information loss errors in which information about the current options is forgotten are unlikely, since option menus will repeat themselves after the final option has been presented. However, if an error did occur due to loss of option information, the error would manifest itself as a choice error, not an information loss error.

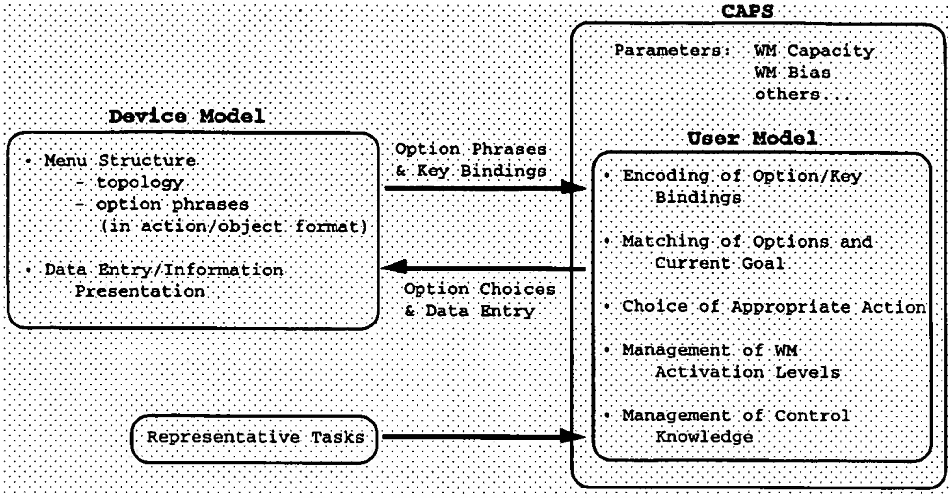
CAPS
Parameters: WM Capacity
WM Bias
others...
Device Model
• Menu Structure
- topology
- option phrases
(in action/object format)
• Data Entry/Information Presentation
Option Phrases & Key Bindings
Option Choices & Data Entry
User Model
• Encoding of Option/Key Bindings
• Matching of Options and Current Goal
• Choice of Appropriate Action
• Management of WM Activation Levels
• Management of Control Knowledge
Representative Tasks

Fig. 1. Components of PBI USER.

Once the User Model has encoded and stored (in WM) a goal task to be achieved (e.g., "add class 247"), the Device Model presents to the User Model the menu options of the simulated interface. As each option of a given menu is presented, the User Model encodes and processes the option information in order to evaluate the appropriateness of the option for the accomplishment of the goal. The result of this processing is the creation of a new symbol in WM representing the "goodness of fit" between the current option and the goal. After the option is evaluated, the only option information maintained in WM is this "goodness of fit" symbol. Depending on the strength of the match between the current option and the goal, the User Model will then either select the current option, eliminate it from consideration, or maintain the option as a potential future choice.

(A maximum of one option can be maintained as a potential future choice at any given moment; since only the "goodness of fit" symbol is maintained, minimal additional WM capacity is consumed.) Thus, in PBI USER total WM load rises as the evaluation of each option is performed, and then falls between option evaluations. After the User Model makes a menu option choice, the Device Model presents options for the next menu of the hierarchy, and the process continues until the goal is achieved or until an information loss or choice error occurs. The User Model also manages the total level of activation and the goal structure of the task (control knowledge). The CAPS architecture includes parameters, such as WM capacity, that will be explained later.

3.1 Factors Addressed through PBI USER

We have used PBI USER to generate a set of hypotheses about the impact of three factors on WM failure: menu structure, WM capacity, and task characteristics.

3.1.1 *Menu Structure.* In any menu-driven computer system, the menus can be structured to emphasize depth or breadth. When depth is emphasized, the number of choices at each menu is minimized, but the number of menus the user must traverse increases. Conversely, broad menus minimize the number of traversed menus, but increase the number of choices at each menu. Depth versus breadth has been the focus of much research in the domain of visual, full-screen menu systems; in general, results indicate that broad and shallow menus are superior to narrow and deep ones, both in terms of accuracy and execution time [Kiger 1984; Parkinson et al. 1988]. In contrast to these results, designers of strictly auditory menus suggest [Engelbeck and Roberts 1989; Pelton 1989; Gould and Boies 1984] that auditory menus should never have more than three options at any given level because of WM limitations, thus forcing a deep and narrow menu structure. However, the top-level menu option labels of a deep and narrow menu structure will tend to be less semantically similar to the structure's terminal options than will the top-level options of a broad and shallow menu structure for the same menu domain. Collins and Quillian's [1969] experiment comparing reaction times in making true-false judgments about assertions concerning concepts (e.g., "Robins eat worms"; "Apples have feathers") showed that greater semantic distance between concepts tends to increase reaction time. Dumais and Landauer [1984] showed that the inclusion of a single ambiguous option label on a menu system degraded subjects' understanding of other option labels. Furthermore, Carpenter and Just [1989] have shown that, in tests of short-term retention of verbal material, the inclusion of a single difficult sentence tends to degrade retention of other sentences presented earlier. Therefore, the traversal of top-level menus in deep and narrow menu structures may create excessive processing loads, increasing the risk of user-generated errors. However, it should be remembered that an alternative explanation for any increase in choice error rates for deep and narrow menu structures is that of inadequate LTM knowledge concerning the relationship between the task at hand and the menu options; this point is revisited in the discussion section.

We have tested PBI USER with two menu structures (PBI-DEEP and PBI-BROAD) that are identical in functionality but differ in topology. The menu hierarchy of PBI-DEEP has four levels of menus, with each menu containing three options (a 3 × 3 × 3 × 3 structure), while the menu hierarchy of PBI-BROAD has two levels, with each menu containing nine options (a 9 × 9 structure). The use of two menu structures allows us to manipulate the *processing* requirements of the PBI task, since the correct option from the top level of PBI-DEEP would be expected to be less closely related to the description of the current task than would be the correct option from the top level of PBI-BROAD. This semantic dissimilarity of top-level options and task descriptions creates heavier demands for processing the top-level options of PBI-DEEP than those of PBI-BROAD.

3.1.2 *WM Capacity.* WM capacity refers to the total activation available for storage and processing. Through the activation capacity constraint of the CAPS architecture, we can directly model the impact of individual differences in WM capacity on WM failure.

3.1.3 *Task Characteristics.* We have also investigated a task characteristic related to task complexity: the number of tasks to be performed sequentially. This characteristic addresses the amount of information that must be stored during the performance of a set of tasks. The number of tasks was manipulated by presenting to PBI USER a request to perform either one single task or a pair of tasks. Through this task characteristic, we were able to manipulate the *storage* requirements of PBI tasks.

An additional task characteristic, task format, was used to control for differences in storage requirements between individual PBI tasks. All tasks performed by PBI USER were given in an action/object/modifier format; for the task "add class 247" the action is "add"; the object is "class"; and the modifier is "247." Differences in task format were generated by changing the type of modifier found in the task. We used three types of modifiers: (1) absent (e.g., "retrieve transcript"), (2) lexical (e.g., "retrieve events **community-service**"), and (3) numeric (e.g., "add class **247**").

3.2 Hypotheses on Information Loss Errors

The PBI USER architecture suggests that it is not the number of options per menu level that determines the magnitude of WM load, but rather the amount of processing and storage required to evaluate the "goodness" of each individual option. Thus WM load should rise, peak, and fall during each option evaluation, and it is the height of each peak relative to total WM capacity that determines the probability of information loss. Due to the higher WM demands for processing the top-level options of PBI-DEEP, we expect the peak WM loads to be higher for PBI-DEEP than for PBI-BROAD. This higher demand for WM processing capacity should result in less WM capacity being left available for the maintenance of stored task information, and therefore it should increase the probability of information being lost. Thus, we have the following:

(H1) PBI USER predicts that *information loss* error rates will be higher for PBI-DEEP than for PBI-BROAD.

Note that H1 challenges the widespread guideline that phone-based interfaces should limit the maximum number of options per menu to three [Engelbeck and Roberts 1989; Gould and Boies 1984; Pelton 1989].

Our model of phone-based interaction predicts WM errors when the demands for activation (due to processing and storage) exceed the activation constraint. Since we know humans have different WM capacities [Daneman and Carpenter 1980], we should expect that users with lower WM capacity should have a greater probability of having their capacity exceeded by the activation demands of phone-based interaction, and therefore these low-capacity WM users should have a greater probability of losing information stored in WM. Thus, we have H2:

(H2) PBI USER predicts that subjects with higher WM capacity would have lower information loss error rates than subjects with lower WM capacity.

For any given level of WM capacity, if we increase the storage requirements for a task (the amount of information that must be maintained in WM to execute the task), then we would expect an increase in the probability of information loss errors (since there would be an increased probability of exceeding the WM capacity constraint).

The two levels of number of tasks provide us with a straightforward means of manipulating the amount of information that must be held in WM during task performance: single tasks should require less storage activation than pairs of tasks.

(H3) PBI USER predicts that increased complexity of tasks (operationalized by number of tasks moving from "single" to "pair") would increase information loss error rates.

3.3 Hypotheses on Choice Errors

Our model of phone-based interaction (PBI USER) depicts menu option choice as the process of comparing each menu option to a description of the current task and then selecting the most appropriate option. This comparison process should be more error prone (due to insufficient knowledge and/or WM constraints) when the menu options are dissimilar to the task descriptions than when the options and task descriptions are closely related, so we expected more choice errors when menu options and task descriptions are dissimilar. In our study, the menu structure PBI-DEEP must categorize all of the functionality of the final 81 terminal nodes into three options at the top-level menu, in contrast to the PBI-BROAD structure which has nine options at the same level. Thus the top-level options of PBI-DEEP are less similar to the task descriptions than the top-level options of PBI-BROAD, leading to higher peak demands for WM processing resources for PBI-DEEP than for PBI-BROAD. Therefore, we have H4:

(H4) PBI USER predicts that *choice* error rates would be higher for PBI-DEEP than for PBI-BROAD.

We expected that the processing requirements for choosing options in the menu structure PBI-DEEP would be greater than for PBI-BROAD. This increased processing requirement would be the result of the additional semantic processing required to disambiguate the option labels in PBI-DEEP. This disambiguation process requires (in PBI USER) the retrieval of information from long-term memory that relates the semantically dissimilar option label to the current task. This additional processing load should result in PBI-DEEP users being more likely than PBI-BROAD users to exceed their WM capacities during the selection of options. On the other hand, the more specific options of PBI-BROAD should require less processing for disambiguation than PBI-DEEP, and therefore we expected a weaker effect between WM capacity and choice error rates for the broad menu structure:

(H5) PBI USER predicts that an interaction effect will occur between WM capacity and menu structure for *choice* error rates. For PBI-DEEP, subjects with higher WM capacity would have lower choice error rates than subjects with lower WM capacity. This effect is predicted to be weaker for PBI-BROAD.

If we increase the storage requirements for a given task, then there would be less activation left available for the processing requirements of menu option choice. Therefore, we expected that as we increase task storage requirements by increasing task complexity (through number of tasks), there would be an increased level of choice errors:

(H6) PBI USER predicts that increased complexity of tasks (operationalized by number of tasks moving from "single" to "pair") would increase choice error rates.

4. METHOD

The purpose of the experiment is to provide empirical evidence for the impact of three factors on WM error rates: (1) structure of the menu hierarchy, (2) individual differences in WM capacity, and (3) WM demands induced by task characteristics.

4.1 Subjects

Eighty-seven students were recruited from local universities and paid U.S. $10.00 to participate in the experiment. Subjects were categorized by their dynamic WM capacity into three groups, based on their scores on the reading span test [Daneman and Carpenter 1980]: low span (20 subjects with scores of 2.0 or 2.5), medium span (45 subjects with scores of 3.0 or 3.5), and high span (22 subjects with scores of 4.0, 4.5, 5.0, 5.5, or 6.0).

4.2 Materials and Apparatus

We have implemented two simulated phone-based interfaces (PBI-DEEP and PBI-BROAD) in the NeXTStep development environment. Figure 2 shows the screen layout used for both of the interfaces. The two systems are functionally equivalent but differ in the topology of their menu hierarchies. The menu hierarchy of PBI-DEEP has four levels of menus, with each menu containing three options (a 3 × 3 × 3 × 3 structure), while the menu hierarchy of PBI-BROAD has two levels, with each menu containing nine options (a 9 × 9 structure). In choosing a functional domain for the PBI systems, our goal was to choose a domain with which all students would be familiar. Toward that goal, we implemented a student information system that involved the domain of student registration/information.

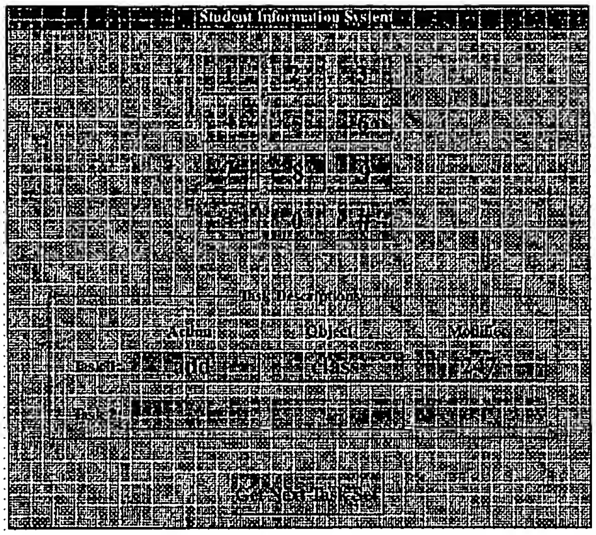

Fig. 2. Keypad and presentation screen.

In addition to choosing a domain that would be familiar to students, we also wanted to choose a domain that would naturally categorize into the 3 × 3 × 3 × 3 structure as well as the 9 × 9 structure. Specifically, since our hypotheses predict that error rates for the PBI-DEEP structure would be higher than those for PBI-BROAD, we did not want to "stack the deck" by choosing a domain that was more appropriate for the 9 × 9 structure than for the 3 × 3 × 3 × 3 structure. With this in mind, we developed the menu structure for PBI-DEEP first and designed it with categories that naturally divided into groups of threes. Thus, the three top-level options of PBI-DEEP were constructed first, followed by suboptions that fit naturally into nested groups of threes (e.g., intramural sports, broken down into men's, women's, and mixed intramurals, and finally broken down into three specific sports). We then arrived at the PBI-BROAD structure by simply using the second- and fourth-level menus from the PBI-DEEP structure.

Phrases for all menu options and feedback pertaining to the PBI systems were prerecorded and stored on the NeXT workstation used in the experiment. Subjects received the recorded options and feedback through headphones, and all subject input was made by clicking on a screen representation of a 12-key touchtone telephone keypad. Task descriptions were presented on the computer screen in an action/object/modifier format as shown in Figure 2. All subject actions (mouse clicks), along with timing data, were captured by the software.

4.3 Experimental Design

In the experiment we wanted to control for the possible effect of unique characteristics of individual tasks on WM error rates. There are three classes of tasks based on number of tasks: (1) single tasks, (2) those tasks that are the first in a pair of tasks, and (3) those tasks that are second in a pair of tasks. These three task classes are referred to as *task order.* In addition to task order, there are another three classes of tasks based on the action/object/modifier description of the tasks: (1) tasks with only action/object (e.g., "retrieve/schedule"), (2) tasks with action/object and a lexical modifier (e.g., "retrieve/social activity information/about concerts"), and (3) tasks with action/object and a numeric modifier (e.g., "add/class/439"). These three task classes are referred to as *task format.* Crossing task order with task format, we obtain nine possible task types.

In addition to the nine task types, we also have the two factors of menu hierarchy structure and dynamic WM capacity. Two menu structures were used (PBI-BROAD and PBI-DEEP), and three levels of dynamic WM capacity were considered (low, medium, and high). Thus, we have a 9 × 2 × 3 design, with task type being a within-subject factor and menu structure and WM capacity being between-subjects factors. Using WM capacity as a blocking variable, subjects were randomly assigned to one of the two menu structures. A total of 27 tasks were presented to each subject, allowing for three replications of each of the nine task types. All subjects were presented with the same 27 tasks, regardless of the menu structure used.

Both of the menu structures used in the experiment (PBI-BROAD and PBI-DEEP) allow for the performance of 81 different tasks, corresponding to the 81 terminal leaves found in each of the two menu structures. From these 81 possible tasks, we selected the 27 tasks that would actually be presented to the subjects, and in the selection of these 27 tasks we balanced a number of experimental design factors. First, three tasks were to be chosen for each of the nine possible task types. Second, the tasks were to be evenly dispersed across the 81 terminal nodes of the menu trees (for example, we did not want all of the tasks to occur in the first few branches of the menu). Third, we wanted to control for a factor referred to as "minimum number of options." We are predicting that the *semantic similarity* of the options and the current task will have an impact on WM load, while the number of options per menu will not. Therefore we wanted to control for the effect of the (minimum) number of options that the user needs to process for making the correct choice. For example, if in the PBI-BROAD structure the current task would be accomplished by choosing the first option at each of the two menu levels, then the minimum number of options for that task would be two. At the other extreme, if the current task would be accomplished by choosing the last option at each of the menu levels of PBI-BROAD, then the minimum number of options would be 18. Similarly, the range of values for the minimum number of options for PBI-DEEP was from four to 12. To control for a minimum number of options, we selected tasks that were evenly dispersed across the range of values for minimum number of options for each menu structure. Finally, within each group of three tasks (for each of the nine task types), we wanted the average value for minimum number of options to be equal to the overall average for the menu structure (for PBI-BROAD this average minimum number of options is 10; for PBI-DEEP this average is eight). To accomplish the selection of the 27 tasks according to these constraints, an integer programming formulation of the task selection problem was implemented in Hyper Lindo [1988] for each of the two menu structures.

In order to complete the design of the experiment, we had to determine (1) the order of presentation of the sets of tasks to the subjects in terms of the sequence of single tasks versus pairs of tasks and (2) the assignment of specific single tasks and pairs of tasks to the sequence order determined in (1). These determinations were made through randomization of the sequence of task sets and randomization of the assignment of specific tasks to the sequence of sets. However, each subject received the tasks in the same order.[2]

4.4 Procedure

Each subject participated in two separate sessions. In the first session, the reading span test was administered to each subject on an individual basis, using the guidelines given in Daneman and Carpenter [1980]. The test requires the subject to read aloud a set of unrelated sentences and then recall the final word from each sentence. Subjects begin with sets of two sentences, followed by sets of three, four, five, or perhaps six sentences. The largest set size for which the subject successfully recalls all of the final words for at least three out of five sets is defined as the subject's reading span. If the subject successfully recalls only two of the five sets, then he or she is assigned a reading span half-way between the current set size and the next lower one. Typical reading span scores range from 2.0 to 6.0, with increments of 0.5. Each span test took approximately 10 minutes to complete.

The second session was the main session of the experiment, in which the subjects interacted with the simulated PBI system. At the beginning of this session, the subjects received verbal instructions on the use of the PBI system and were then given three sets of practice tasks to perform (two single tasks and one pair of tasks). After all subjects had completed the practice tasks, and all initial questions had been answered, the main part of the session began. Each subject performed a total of 27 tasks with a single menu structure (either PBI-BROAD or PBI-DEEP). Eighteen of the tasks were presented in the form of nine pairs, and the other nine tasks were presented singly. Subjects were instructed to perform tasks in the order of presentation (for those tasks that were presented in pairs). The subjects interacted with the PBI systems solely by listening to auditory options over headphones and using the computer's mouse to click on the screen. A representation of a standard 12-key telephone keypad was displayed on the screen, and subjects clicked on the keypad to make option choices.

Task presentation was also done on the screen, using the layout shown in Figure 2. When ready to begin a new task, the subject used the computer's

[2] For details of the experimental design and materials, see Huguenard [1993].

mouse to click the button "Get Next Task Set." This caused a description of the next set of tasks to be displayed on the screen, where a set consisted of either a single task or a pair of tasks. In the case of a pair of tasks, the tasks were presented sequentially, with the first task being erased before the second task was displayed. The task descriptions were presented in slots labeled for actions, objects, and modifiers, as shown in Figure 2. If a task had no modifier, then the modifier slot was blank. The task description for the first task was displayed for a predetermined duration (based on an average reading speed), and then the action/object/modifier slots were erased. At this point, if a pair of tasks was being presented then the second task would be displayed in the "Task 2" set of slots. After all tasks had been presented the PBI began presenting auditory options, and the user began performing the tasks. If at any time the subject forgot any task information, then the forgotten information could be retrieved by clicking on the corresponding action/object/modifier slot. For instance, if the subject remembered that the first task involved adding a class, but could not remember the class number, then he or she could click on the "Modifier" slot of "Task 1," and the class number would be displayed for a predetermined length of time. By clicking on one of the action/object/modifier slots, the subject would be indicating that an information loss error had occurred (specifically, a *rehearsal* error, as discussed in Section 5.1). If at any time the subject realized an incorrect option choice had been made, the incorrect option choice could be "undone" by clicking on the asterisk (*) button. Clicking on the asterisk button would return the subject to the next-higher menu in the hierarchy and could be done repeatedly if needed. By clicking on the asterisk button, the subject would be indicating that a choice error had occurred (specifically, a *navigational* error, as is discussed below).

The experimenter was available during the entire session to answer questions about the experiment. After completing all tasks in a set, the subject would click the button "Get Next Task Set," and the process began again. The time required to complete this session was approximately 50 minutes.

5. DATA ANALYSIS

5.1 Classification of Responses

Our theoretical classification of error types was operationalized in the following manner: information loss errors were operationalized as *rehearsal errors* or *task failure errors,* while choice errors were operationalized as *navigational errors.*

For information loss, rehearsal errors were said to occur when the subject forgot a parameter of a task such as a modifier (e.g., add class **247**) or the object of the task (add **class** 247), but after making a request to review this parameter, the subject completed the task successfully (subjects were allowed only *one* such request per task). As an example of a rehearsal error, consider a subject interacting with the PBI-DEEP system who is to perform the single task "Add Class 247," but forgets the number of the class to be added (a rehearsal error involving the task modifier). The following dialog describes the options presented by the PBI-DEEP system and the corresponding actions chosen by the subject leading up to the rehearsal error:

Options Presented: "To work with Academic Information, Press 1;
To work with Extracurricular Activities, Press 2;
To work with Personal Information, Press 3."

Subject's Response: 1 (Work with Academic Information)

Options Presented: "To work with Classes, Press 1;
To retrieve Class Assignments, Press 2;
To retrieve Academic Records, Press 3."

Subject's Response: 1 (Work with Classes)

Options Presented: "To work with Schedule, Press 1;
To work with Class Information, Press 2;
To work with Instructor Information, Press 3."

Subject's Response: 1 (Work with Schedule)

Options Presented: "To add a class, Press 1;
To drop a class, Press 2;
To retrieve your schedule, Press 3."

Subject's Response: 1 (Add a Class)

Options Presented: "Please enter the number of the class you wish to add."

Subject's Response: At this stage the subject realizes he or she has forgotten the class number, and so presses the "Modifier" button for task 1 (see Figure 2). Pressing the Modifier button results in the momentary display of the modifier value, after which the subject can complete the task. Pressing the Modifier button also allows the software to log when the rehearsal error occurred and which task parameter was involved.

The second type of information loss, task failure, was said to occur when the subject failed to complete the task correctly (by failing to complete the task or by completing the wrong task). Task failure may occur even after a request for a forgotten parameter has been made, and in such a situation the error would be counted as a task failure, not a rehearsal error. In our experimental setting, rehearsal errors were a recoverable type of information loss since the subject was able to complete the task correctly; on the other hand, we consider task failure to be the result of a more severe type of information loss, since we assume subjects have forgotten more than one task parameter. However, since both are forms of information loss, we expected hypotheses H1–H3 to hold for both rehearsal and task failure errors. As an example of a task failure error, consider the same dialog shown above, but suppose that in addition to forgetting the class number (task modifier), the subject also forgets whether the class is to be added or dropped (task action). The subject could press the Modifier or the Action button, but not both, and so would not be able to retrieve all the information necessary to complete the task. At best, the subject could retrieve the modifier value and make a guess on the value of the task action. A task failure would be counted if the subject either (1) gives up and does not complete the task at all or (2) guesses incorrectly, thereby completing the wrong task (e.g., drops instead of adds the course).

For choice errors, navigational errors were said to occur when the subject made at least one incorrect option choice while traversing the menu structure but completed the task correctly by backtracking. Navigational errors could be the result of (1) WM capacity being insufficient to satisfy the processing requirements of the task and/or (2) lack of knowledge about how the current task relates to the options in the current menu. We partially controlled for knowledge by randomly assigning subjects to experimental conditions. However, since we could not completely control for knowledge through randomization, we chose a task domain (student affairs) with which our subjects (college students) were familiar and for which individual differences in knowledge should not have been significant. Therefore, we expected WM limitations to be an important factor on the frequency of navigational errors, and we expected hypotheses H4–H6 to hold for navigational errors. As an example of a navigational error, consider a subject interacting with the PBI-DEEP system who is to perform the single task "Add Class 247." This subject makes an incorrect option choice while navigating through the menu hierarchy, but then "backs up" a menu level to correct the mistake and then complete the task successfully. The following dialog describes the options presented by the PBI-DEEP system and the corresponding actions chosen by the subject leading up to the navigational error:

Options Presented:	"To work with Academic Information, Press 1; To work with Extracurricular Activities, Press 2; To work with Personal Information, Press 3."
Subject's Response:	1 (Work with Academic Information)

Options Presented:	"To work with Classes, Press 1;
	To retrieve Class Assignments, press 2;
	To Retrieve Academic Records, Press 3."

Subject's Response:	1 (Work with Classes)

Options Presented:	"To work with Schedule, Press 1;
	To work with Class Information, Press 2;
	To work with Instructor Information, Press 3."

Subject's Response:	Makes a *navigational error* by choosing option 2 (Work with Class Information)
Options Presented:	"To check prerequisites, Press 1; To check class time and location, Press 2; To check for an open class, Press 3."
Subject's Response:	At this point the subject realizes that none of the presented options are relevant to the task at hand, and so he or she backs up a menu level by pressing the * key (see Figure 2).

Options Presented:	"To work with Schedule, Press 1;
	To work with Class Information, Press 2;
	To work with Instructor Information, Press 3."

Subject's Response:	Subject now makes the correct choice of option 1 (Work with Schedule) and goes on to complete the task without incident.

Given these operationalizations of the error types, we categorized the possible responses for a given task into five groups: (1) completed correctly with no errors, (2) task failure, (3) completed with rehearsal error only, (4) completed with navigational error only, and (5) completed with both navigational and rehearsal errors. Using these five response categories, task failure errors are represented by response (2); rehearsal errors are represented by the combination of responses (3) and (5); and navigational errors are represented by the combination of responses (4) and (5).

5.2 Statistical Analysis

Our data consist of 27 repeated discrete measures from each of 87 subjects, observed under experimental conditions controlling both within- and between-subjects factors. The within-subjects factors are task format (3 levels) and task order (3 levels). The between-subjects factors are menu type (2 levels) and reading span (3 levels). If we assume that all variability in responses attributable to subjects is accounted for by reading span and menu type, then responses to different tasks will be independent given those factors. Under this assumption a multivariate logistic regression model would be appropriate, and an analysis of deviance could be carried out using any of a number of statistical computing packages (e.g., PROC LOGIST from SAS [SAS Institute 1987]). This independence assumption is untenable however, since individual solution strategies, differences within reading span group, and other (unobserved) subject covariates are likely to induce dependence between tasks within subjects beyond that attributable to experimental conditions.

Statistical models for repeated discrete measures that account for within-subject, between-task dependence are common in the educational testing literature (e.g., Longford [1995]). In testing settings however, interest lies most commonly in characteristics of individual subjects and/or test questions, and the statistical methodology is designed for very large subject samples (500 or 1000 subjects is not uncommon). Hence, available statistical software (e.g., MULTILOG [Thissen 1986]) is not well suited to the estimation of experimental effects in our setting. Thus, we have developed statistical models and estimation programs from first principals (as in Carlin et al. [1992]). For the current data, we have developed a statistical model for the probability P_{iml} that subject i makes response l to task m. In the Appendix we present this statistical model and the essence of its development; all other details are described elsewhere [Patz et al. 1996]. This model contains the multivariate logistic regression model discussed in the previous paragraphs, as a special case. Patz et al. [1996] demonstrate that ignoring residual within-subject between-task dependence leads to an exaggeration of between-subjects effects. We avoid this error by appropriately modeling the dependence structure of our data.

In our formal analysis of the experimental data, we assessed the effects of task factors, menu type, and reading span group by comparing pairs of nested models of the form (3) (see the Appendix). Models were compared with likelihood-ratio statistics, which compare the likelihood of the data under one model with the likelihood of the data under an alternative model on a log scale (see Agresti [1990, pp. 48–49]; see also Haberman [1977] and Koehler and Larntz [1980]). Likelihood-ratio statistics are also called model deviances in the loglinear modeling and logistic regression literature (see Agresti [1990, p. 83]). These statistics are approximately chi-squared distributed with degrees of freedom equal to the difference in the number of free parameters in the two models when the smaller model is correct.

In addition, some of the hypotheses in Section 3 assert that certain response rates, such as rehearsal error rates, should be ordered by a design factor, such as reading span category. A clearer picture of these hypotheses can be gained considering a one-sided test for the hypothesis that the rate predicted to be highest, minus the rate predicted to be lowest, was indeed positive (there was insufficient power, given the sample size, to do more detailed comparisons). To construct the test statistic z, we obtain maximum-likelihood estimates of each rate using the final fitted model and divide the difference in these estimated rates by the appropriate standard error. In the results section this test will be referred to as a "high-low contrast."

The chi-squared approximation to the likelihood-ratio (LR) test (reported as model deviances) and the normal approximation to the high-low contrasts (reported as z statistics) are often not very accurate in small-sample problems. Many of the effects we are interested in generated test statistics so far from the usual "significance" levels that the presence or absence of each effect is not in question. For other cases, we were able to assess the accuracy of these approximations by analyzing model fits to simulated data deviating in various ways from the actual data obtained in the experiment. We will return to the assessment of the robustness of our data analysis after the results are reported.

Most of the results we report should be viewed in the context of the Bonferonni correction for simultaneous tests [Morrison 1990, pp. 32–33]. For the reader's convenience we supply the appropriate significance statements here. There were 15 high-low contrasts of interest in the experiment; therefore cutoffs for the high-low contrasts (z) are corrected for 15 simultaneous tests and are 2.71, 3.21, and 3.82 for nominal levels $\alpha = 0.05$, 0.01, and 0.001, respectively. The cutoffs for the chi-squared likelihood ratio tests are corrected for seven simultaneous tests, corresponding to the "main effects" of each of the four task factors, and interactions of menu structure with each of the other three factors. The corrected cutoffs are shown in Table I.

6. RESULTS

Results are presented first for the overall effects of menu structure, WM capacity, and task characteristics. Next the specific results that support (or fail to support) the research hypotheses are presented, followed by a discussion concerning the robustness of the results. We then conclude this

Table I. Corrected Cutoffs for the Chi-Squared LR Tests

	Nominal α		
df	0.05	0.01	0.001
8	21.0	25.2	31.0
4	14.1	17.7	22.7
2	9.9	13.1	17.7
1	7.2	10.2	14.5

section with the presentation of a few qualitative analyses that further test our interpretation of the quantitative results.

6.1 Overall Results

Table II shows the observed rates for the five response categories. Table III shows the contribution of each experimental factor to the model discussed in Section 5 in which the dependent variable is a five-element vector corresponding to the five response categories. Menu structure had a significant effect on overall response rates (LR of 39 on 4 d.f. $p < 0.001$). The rates shown in Table II suggest that the impact of menu structure is mainly on navigational errors (17.23% for PBI-DEEP, 7.74% for PBI-BROAD).

Table III shows that WM capacity was significant at the 0.05 level (using the conservative 0.05 cutoff point of 21.0 for eight degrees of freedom). As an example of the impact of WM capacity on response rates, the observed rates for the response "no errors" were 71.21% for the high WM capacity subjects, 65.93% for the medium, and 59.81% for the low (combined for both menu structures).

Table III also shows that task characteristics (task order and task format) were significant. The impact of task order on response rates is exemplified by the observed rates for the "no errors" response (combined for both menu structures): 79.44% for single tasks and 59.07% for paired tasks. Task format resulted in the observed "no errors" rates of 73.18% for no modifier, 67.94% for lexical modifier, and 56.45% for numeric modifier (again, combined for both menu structures).

Finally, Table III shows that the interaction between menu structure and task format is also significant. This interaction is explained in the presentation of the results for choice errors. The other two interactions were not significant (i.e., menu structure × WM and menu structure × task order).

6.2 Implications of Results for Research Hypotheses

In order for task responses to be classified as rehearsal errors or navigational errors, tasks are required to be completed correctly. Any response classified as task failure would not have the opportunity to be interpreted as either a rehearsal or navigational error. Therefore when calculating error rates for rehearsal and navigational errors, we do not include task failures in the denominator; in other words, we calculated rehearsal and

Table II. Overall Response Rates by Menu Structure

Response	PBI-DEEP	PBI-BROAD
no errors	61.41%	70.20%
task failure	10.08%	10.19%
correct w/rehearsal	7.32%	8.25%
correct w/navigational	**17.23%**	**7.74%**
correct w/rehearsal & navigational	3.96%	3.62%

Table III. Assessment of Experimental Factors in the Seven-Factor Statistical Model

Factor	Likelihood Ratio	d.f.	p
menu structure	39	4	<0.001
WM	22	8	<0.05
task format	187	8	<0.001
task order	225	8	<0.001
menu structure × task format	29	8	<0.01

navigational error rates only for tasks which were eventually completed correctly.

6.2.1 *Information Loss Hypotheses*

(H1) *Information loss error rates higher for PBI-DEEP than for PBI-BROAD:* No significant differences were found between the two menu structures in terms of rehearsal error rates (failing to support H1), with PBI-DEEP and PBI-BROAD having observed rehearsal error rates of 12.55% and 13.21%, respectively (high-low contrast $z = 0.48$, not significant (ns)). In addition, task failure rates for the two menu structures did not differ significantly (LR of 1.0 on 1 d.f., ns),[3] with PBI-DEEP and PBI-BROAD having observed task failure rates of 10.08% and 10.19%, respectively. Since we consider the primary cause of task failure to be information loss, this task failure result also fails to support H1.

(H2) *Information loss rates lower for higher-capacity WM subjects than low WM capacity subjects:* As WM capacity decreased from high to low, observed rehearsal error rates increased for both menu structures, as predicted in H2, but this effect did not reach statistical significance (high-low contrast $z = 1.67$, ns, using the conservative Bonferroni correction for multiple comparisons). The three levels of WM capacity (high, medium, and low) resulted in observed rehearsal error rates (combined for both menu structures) of 11.61%, 12.23%, and 15.89%. On the other hand, WM level did have a significant effect on observed *task failure* rates for both menu structures (lending support to H2).

[3] Likelihood ratio (LR) statistics (instead of high-low contrasts, as in Rehearsal and Navigational errors) were used to test Task Failure effects. This could be done because the Task Failure rates are not conditional on any other type of error, while the rates for Rehearsal and Navigational errors are conditional on Task Failure having not occurred.

Table IV. Observed Navigational Error Rates by WM and Menu Structure

	Navigational Error Rates	
WM	PBI-DEEP	PBI-BROAD
high	19.93%	12.68%
medium	22.91%	11.81%
low	29.44%	14.52%

The three levels of WM capacity (high, medium, and low) resulted in observed task failure rates (combined for the two menu structures) of 5.73%, 11.19%, and 12.59% (LR of 14.0 on 2 d.f., $p < 0.01$).

(H3) *Information loss rates lower for single tasks than for pairs of tasks:* The task complexity feature of number of tasks had a significant effect on the frequency of rehearsal error rates (supporting H3). Combining rehearsal error rates for the two menu structures, number of tasks (single task versus paired task) resulted in observed rehearsal error rates of 3.05% and 18.34% (high-low contrast $z = 12.07$, $p < 0.001$). In addition, the number-of-tasks factor had a significant effect on the frequency of task failure rates for both menu structures (again providing support for hypothesis H3). For instance, number of tasks (single task, paired task) resulted in observed task failure rates (combined for menu structure) of 3.83% and 13.28% (LR of 127.0 on 1 d.f., $p < 0.001$).

In summary the results failed to support H1; H2 was supported by task failure rates, but not by rehearsal error rates; and H3 was supported by both types of information loss errors. That is, in terms of information loss there is no difference between menu structures, a significant difference among WM capacity groups on task failure error rates, and a significant impact of number of tasks. While the observed rehearsal error rates did increase as WM decreased (as hypothesized in H2), this effect was not statistically significant.

6.2.2 *Choice Error Hypotheses*

(H4) *Choice error rates higher for PBI-DEEP than for PBI-BROAD:* Menu structure had a significant effect on navigational error rates (supporting H4), with observed error rates of 23.56% for PBI-DEEP and 12.65% for PBI-BROAD (high-low contrast $z = 4.76$, $p < 0.001$). Accordingly, results for H5 are presented separately for each menu structure.

(H5) *Interaction between WM capacity and menu structure for choice error rates:* Contrary to the prediction of H5, WM level did not have a significant effect on the frequency of observed navigational error rates for either PBI-DEEP (high-low contrast $z = 1.40$, ns) or PBI-BROAD (high-low contrast $z = 0.61$, ns). Referring to Table IV, we see that as WM level decreased from high to low, observed navigational error rates did increase by nearly 10 percentage points for PBI-DEEP,

Table V. Observed Navigational Error Rates by Number of Tasks and Menu Structure

	Navigational Error Rates	
Number of Tasks	PBI-DEEP	PBI-BROAD
single	22.13%	8.47%
pair	24.37%	14.95%

while error rates for PBI-BROAD increased by less than three percentage points; however, this effect was not strong enough to provide support for the interaction predicted by H5.

(H6) *Choice error rates lower for single tasks than for pairs of tasks:* Table V gives the observed navigational error rates for each level of number of tasks. Number of tasks had a significant effect on navigational error rates for PBI-BROAD (high-low contrast $z = 3.42$, $p < 0.01$), but not for PBI-DEEP (high-low contrast $z = 0.87$, ns). This result provides partial support for H6.

In summary, hypothesis H4 was supported by the results of navigational error rates; hypothesis H6 was supported for the menu structure PBI-BROAD, but not for PBI-DEEP; and hypothesis H5 was not supported. That is, in terms of navigational error rates, there is a significant difference between menu structures, a significant impact of number of tasks for PBI-BROAD (but not for PBI-DEEP), and no significant interaction between WM capacity and menu structure.

6.3 Robustness of Results

As indicated in Section 5.2 (Statistical Analysis), at the relatively small (for statistical purposes) sample size of our experiment, it is important to ask whether the results are stable when the data are perturbed.

To answer this question, we performed a Monte Carlo simulation study. Using our fitted model, we simulated data from $N = 87$ subjects, perturbed the fitted values several times, and performed a representative subset of our likelihood-ratio tests. Across 10 Monte Carlo replications from the fitted model, we were able to assess the size of each of the main task factor effects. If the model were very sensitive to perturbations in the data, then we should expect to see wide fluctuations in the task factor effect sizes from replication to replication. The results, summarized in Table VI, suggest that effect sizes are quite stable across replications, and hence our conclusions about the presence or absence of various factor effects are reasonably robust against perturbations in the data. Referring to Table VI, we can see that inferences based on the original data are not overly sensitive to small changes in the data; effects clearly present in the original data remain so in the simulated data sets. Results presented in Patz et al. [1996] show similar stability of effect sizes for the interaction terms Menu × Navigational Error and Menu × Rehearsal Error.

Table VI. Stability of Main Task Factor Effects with Simulated Data

Data Source	BASE -2*LOG Likelihood	Overall Task Factor Effects			
		Menu	WM	Modt	Tord
Experiment	−2250.0	39	22	187	225
Sim.					
1	−2218.0	37.2	21.6	199.6	228.3
2	−2190.0	24.5	42.4	199.6	216.5
3	−2276.4	34.5	34.0	213.2	213.7
4	−2257.8	60.7	25.8	192.9	228.3
5	−2354.8	21.8	24.7	163.2	169.7
6	−2222.9	36.8	29.4	164.5	220.7
7	−2336.2	43.6	41.4	165.3	226.5
8	−2333.4	51.4	26.9	206.9	233.8
9	−2267.5	47.3	18.4	198.9	248.6
10	−2257.9	40.2	21.2	289.9	196.5
Summary					
Mean	−2271.5	39.8	28.6	199.4	218.3
Median	−2262.7	38.7	26.4	199.3	223.6
Stdev	55.0	11.8	8.3	36.8	21.8
Min	−2354.8	21.8	18.4	163.2	169.7
Max	−2190.0	60.7	42.4	289.9	248.6

6.4 Qualitative Results

Qualitative analyses were performed for rehearsal and navigational errors in order to check for differences between PBI-DEEP and PBI-BROAD that would not be captured by our quantitative analyses. These qualitative analyses also served to further test our interpretation of the quantitative results.

6.4.1 *Qualitative Results for Information Loss Errors.* A qualitative analysis of the rehearsal errors showed that in addition to the similarity of PBI-DEEP and PBI-BROAD in overall rehearsal error rates (7.32% for PBI-DEEP; 8.25% for PBI-BROAD), there were no differences between the two menu structures in terms of which task parameters were forgotten (see Table VII). This is further evidence that there is no difference in information loss error rates between the two menu structures.

6.4.2 *Qualitative Results for Choice Errors.* A qualitative analysis of navigational errors was performed for data of 40 subjects (20 per menu structure). We classified the errors for the 40 subjects according to the "level" at which they were initiated. By level of initiation we mean the level within the menu hierarchy (one or two for PBI-BROAD; one, two, three, or four for PBI-DEEP) at which a navigational error was begun. Table VIII gives the results of this categorization, and as expected, most navigational errors were initiated at the upper levels of both menu structures, where option labels tend to have less in common with the task description. In

Table VII. Rehearsal Errors Categorized by Forgotten Task Parameter

Menu	Modifier	Object	Action
PBI-DEEP	54.96%	38.17%	6.87%
PBI-BROAD	53.90%	39.72%	6.38%

Table VIII. Navigational Error Rates and Proportions Categorized by Menu Level of Initiation

	Menu Level			
Menu	1	2	3	4
PBI-DEEP error rate:	12.53%	6.87%	4.65%	0.81%
PBI-BROAD error rate:	9.18%	5.39%	n/a	n/a

addition, Table VIII shows that navigational error rates were strictly decreasing with menu level.

We then wanted to compare the navigational error rates in Table VIII between the two menu structures. However, one must be careful in performing such comparisons to ensure that the menu level comparisons being considered are meaningful. For instance, it would not be meaningful to compare level 1 of PBI-DEEP with level 1 of PBI-BROAD. On the other hand, the options of level 2 in PBI-DEEP are identical to those found in level 1 of PBI-BROAD, so by combining[4] the navigational error rates for levels 1 and 2 of PBI-DEEP, we obtain an error rate representative of the same navigational progress as that of level 1 in PBI-BROAD. Similarly, by combining the navigational error rates for levels 3 and 4 of PBI-DEEP, we obtain an error rate appropriate for comparison with level 2 of PBI-BROAD. The resulting navigational error rates for equivalent menu level 1 were 18.53% and 9.18% for PBI-DEEP and PBI-BROAD, respectively, and for equivalent menu level 2 were 5.42% and 5.39%. Thus, observed navigational error rates were higher at the upper levels (equivalent menu level 1) of PBI-DEEP than for the equivalent level of PBI-BROAD. On the other hand, error rates at the lower levels that tend to be more closely related to task descriptions (equivalent menu level 2) are similar between the two menu structures.

7. DISCUSSION

In this section we first discuss the results related to H2 and H3 which support our predictions concerning WM failure due to *storage* demands. Then we discuss the results for H1, H4, and H5 together because jointly

[4] Probabilities were combined multiplicatively in the following fashion. Let p_1 represent the probability of initiating a navigational error at level 1 of PBI-DEEP and p_2 represent the probability of an error initiation at level 2. The combined probability of initiating a navigational error at either level 1 or level 2 = $[1 - ((1 - p_1)(1 - p_2))]$.

Table IX. Summary of Support for Hypothesis by Error Type

	Supported by Results?		
Hypothesis	Rehearsal	Task Failure	Navigational
Information Loss Errors			
H1 (higher for PBI-DEEP than BROAD)	no	no	n/a
H2 (lower for high WM capacity)	no	yes	n/a
H3 (lower for single tasks than pairs)	yes	yes	n/a
Choice Errors			
H4 (higher for PBI-DEEP than BROAD)	n/a	n/a	yes
H5 (WMxMenu Structure interaction)	n/a	n/a	no
H6 (lower for single tasks than pairs)	n/a	n/a	yes (BROAD) no (DEEP)

they seem to indicate that PBI-DEEP does not generate higher *processing* loads than PBI-BROAD. Our interpretation of the results for H1, H4, and H5 is then further refined by looking at the results for H6. A summary of the support found for all the hypotheses is given in Table IX.

Hypothesis H2, which predicted an increase in information loss error rates as the WM level decreased, was not supported by the results for rehearsal errors, but was supported by the task failure results. While observed rehearsal error rates did increase as the WM level decreased, this effect was not strong enough to reach statistical significance. One explanation for this lack of significance could be our relatively small sample size, coupled with the conservative multiple comparison correction we used, but we offer an additional explanation. Consider the distribution (Figure 3) of WM load over all the tasks given in the experiment, in which most of the tasks have low to moderate WM load and in which the proportion of tasks decreases as WM load increases.

In Figure 3 two "bands" of WM capacity are represented: one extending from "a" to "b" for the low-span subjects and one extending from "c" to "d" for the high-span subjects. These bands indicate the relationships between WM capacity, WM load of tasks, and information loss errors. For instance, the low-span subjects would generate (1) no information loss errors for tasks with WM load less than "a," (2) rehearsal errors for tasks with WM load between "a" and "b," and (3) task failure errors for tasks with load greater than "b." Similarly, high-span subjects would generate (1) no information loss errors for tasks with WM load less than "c," (2) rehearsal errors for tasks with WM load between "c" and "d," and (3) task failure errors for tasks with WM load greater than "d."

Notice that although the area representing rehearsal errors for the high-capacity WM group is smaller than that for the low-capacity WM group, the difference is not great. However, if we consider the area representing task failure for the low-capacity group (all tasks beyond "b"), we note that it is much greater than the corresponding area for the high-capacity group (all tasks beyond "d"). Thus, although the high-capacity subjects do have lower error rates than the low-capacity subjects for both rehearsal and task failure errors, it is easier to distinguish between

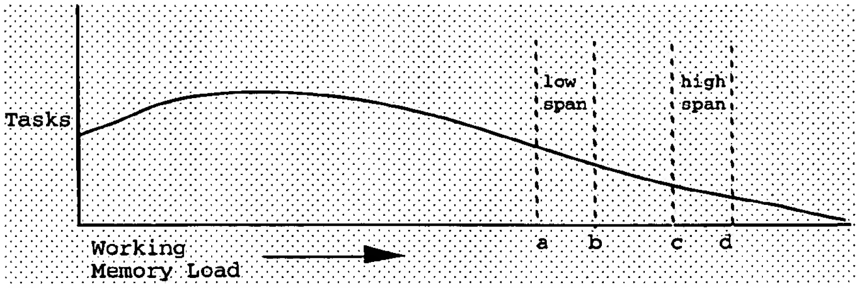

Fig. 3. Relationship of WM capacity groups to a distribution of tasks by WM load.

the task failure error rates. Therefore, while there was not a significant effect of WM on rehearsal error rates, we believe that the true test of the importance of WM as a predictor of information loss error rates is with the impact of WM capacity on task failure rates.

The number-of-tasks feature had a significant effect on information loss error rates (rehearsal and task failure errors), as predicted by H3. The explanation of this result is straightforward: as we increase the number of tasks, we increase the total number of items to be stored in WM, and this increases the likelihood of information loss errors.

In H1 we proposed that PBI-DEEP would generate higher WM loads due to the processing of options labels at the top level and that this would result in higher information loss error rates for PBI-DEEP. Both rehearsal and task failure error rates failed to support this hypothesis (that is, error rates were not significantly different between PBI-DEEP and PBI-BROAD). There are three alternative explanations for these results:

(a) PBI-DEEP *does not* generate higher processing demands than PBI-BROAD.
(b) PBI-DEEP *does* generate higher processing demands than PBI-BROAD, but these higher processing demands come at the expense of higher navigational error rates, instead of higher information loss error rates.
(c) PBI-DEEP *does* generate higher processing demands than PBI-BROAD, but higher processing demands do not increase the likelihood of WM failure.

Our proposed explanation is a refinement of explanation (b). The results failing to support H1 do not help us to discriminate among these alternatives, but by integrating the results for hypotheses H1, H4, H5, and H6 we can eliminate one of the explanations and propose an underlying mechanism that explains the results for these four hypotheses.

The results for H4 show that navigational error rates are higher for PBI-DEEP than for PBI-BROAD. This specific result may be due to insufficient long-term memory (LTM) knowledge and/or higher processing loads for disambiguating the top-level options. In PBI USER we model disambiguation as the retrieval of disambiguating information from LTM, the *temporary* storage of this information in WM, and the actual processing, which relates the disambiguating information to the option labels. Note that as soon as the processing of a label is completed, the disambiguating information being held in WM is no longer needed. Thus, our definition of processing includes demands for storage, but these storage demands are not for the ongoing storage of task parameters, but rather are for the temporary storage of disambiguating information retrieved from LTM. Unfortunately, the results for H4 in isolation cannot tell us if the higher proportion of navigational error for PBI-DEEP is due to a lack of LTM knowledge or to the processing of this knowledge or to both.

Moving on to H5, we predicted that subjects with more capacity will make fewer navigational errors than subjects with less capacity and that this effect would be more pronounced in PBI-DEEP. The presence of this interaction effect between menu and WM capacity would indicate that PBI-DEEP indeed generates higher processing demands, given the results in H1 and H4. If this interaction effect does not exist, this would indicate that the difference in navigational error rates found in H4 is due ONLY to insufficient LTM knowledge. The predicted interaction was not statistically significant. However, the observed error rates were in the right direction, that is, navigational error rates for PBI-DEEP did increase by nearly 10 percentage points as WM decreased from high to low, while navigational error rates for PBI-BROAD increased by less than three percentage points in the same situation. Consequently, the statistical analysis supports our first explanation (explanation (a): PBI-DEEP does not generate higher processing loads). But the observed error rates leave open the possibility that there is a higher processing load for PBI-DEEP (explanation (b)), but the effect is too small to reach significance in our experimental design (i.e., due to a between-subjects design for both menu structure and WM capacity factors and to the relatively small sample size). We will return to this issue after discussing H6.

The last explanation (explanation (c)) states that higher processing demands do not increase the likelihood of WM failure. This explanation would repudiate the current theoretical view of WM as a system for the temporary storage *and* processing of information. H6 predicted that navigational error rates would increase as the number of tasks varied from one to two. The results show that navigational error rates increased for PBI-BROAD (as predicted), but not for PBI-DEEP. Explanation (c) is not supported by the results for PBI-BROAD. These results suggest that subjects in PBI-BROAD were able to use LTM knowledge to disambiguate top-level options more successfully when they performed single tasks than when they performed paired tasks, because in the case of single tasks they have spare capacity for doing some extra processing (this assumes that LTM knowledge and disambiguation demands are equivalent for single and paired tasks; we used an integer programming approach to design our experimental materials, as explained in the method section, with this objective in mind). On the other hand, why were PBI-DEEP subjects not able (or willing) to use their spare capacity for disambiguating top-level options in single tasks? To answer this question, we propose that PBI-

DEEP users have a different strategy for selecting options than PBI-BROAD users. This is a strategy concerning how to make decisions. We propose that PBI-DEEP users decide to exert less processing effort because the potential benefit for doing it is not worth the effort. This proposed mechanism explains the pattern of results for H1, H4, H5, and H6. We present our explanation in two different ways: in terms of the strategic choice on how to decide and in terms of the revised version of PBI USER.

We assume that PBI-DEEP generates higher processing demands than PBI-BROAD and that these demands increase navigational error rates (explanation (b)), but not information loss error rates (as confirmed by H1 results). We propose that the difference in navigational errors between the two menus (in H4) is the result of *both* lack of LTM knowledge and higher processing demands. We propose that PBI-DEEP subjects avoid exerting the full effort required for disambiguation by making a strategic choice (a metadecision) of choosing the first option with a reasonably high level of fit with the task and that the required level of fit is lower than that used for PBI-BROAD. If this is true, then we can explain the weak interaction effect between menu structure and WM capacity (H5).

The strategic-choice explanation is based on prior research [Payne et al. 1988; 1990; 1993] in metadecision making which has shown that decision makers make a tradeoff between effort and benefit when making decisions on how to decide, based on factors such as the number of options. When subjects make an incorrect option choice in PBI-DEEP, they have to listen to at most the three options in the next menu level before realizing their mistake (and in fact, for navigational errors, subjects do not progress further than one menu level after an incorrect decision). This is in contrast to PBI-BROAD, where it is worth the expense of fully disambiguating option labels because an incorrect option choice may lead to the tedious task of listening to nine more options before detecting the error. Note that navigational errors were operationalized in our empirical work as those tasks completed successfully after having made at least one incorrect option choice. This strategic choice can explain the results in H6 where it seems that PBI-BROAD subjects are capable (and willing) to spend their spare resources doing processing of option labels while executing single tasks, while PBI-DEEP subjects are not. The result is that PBI-BROAD subjects make less navigational errors in PBI-BROAD for single tasks versus paired tasks while PBI-DEEP subjects do not.

We now describe how the revised version of PBI USER simulates different strategies for option choice and how these strategies explain the differences in the results between PBI-BROAD and PBI-DEEP for hypotheses H4, H5, and H6. The first step in simulating phone-based interaction in PBI USER involves the encoding of the task in terms of the task action, object, and (for some tasks) modifier (e.g., "add/class/247"). Then the first option of the top-level menu is presented, and PBI USER encodes the corresponding action, object, and (for some options) modifier (e.g., "work with/academic information"). PBI USER then performs a matching between the encodings of the task and the option, with the result being the creation

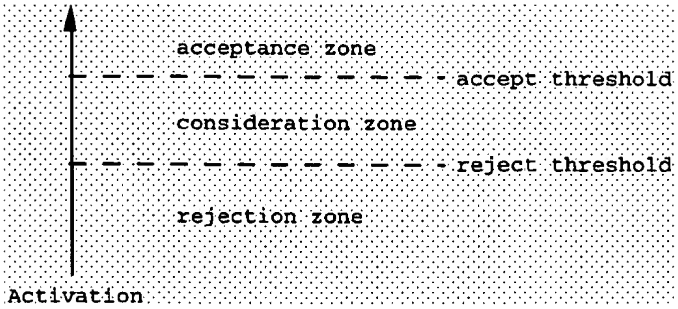

Fig. 4. Activation thresholds for accepting and rejecting options.

of a new symbol in WM representing the "goodness of fit" between the current option and the task. The better the match between the task and the option, the higher the level of activation given to the goodness-of-fit symbol, while a poor task/option match will result in a low level of activation. The strength of activation associated with a given goodness-of-fit symbol determines what action PBI USER will take concerning the corresponding option.

Productions in PBI USER have two *activation thresholds* that serve as trigger points for rejecting or choosing a given option. The lower of the two thresholds is called the reject threshold, and the higher is called the accept threshold (Figure 4). If the activation level of a goodness-of-fit symbol is below the reject threshold, then that option will be immediately rejected from consideration, and processing will begin for the next option. If the goodness-of-fit activation is above the accept threshold, then the corresponding option will immediately be chosen, and any further option presentation for the current menu will be halted. Finally, if the goodness-of-fit activation level is between the two thresholds, then the current option will be kept as a possible future choice (it may be chosen if no superior match is found for other options), and processing will begin for the next option. This use of activation-based thresholds is similar to the concept of criterion-based selection of menu alternatives proposed by MacGregor et al. [1986].

The matching of an option to a task is performed as a two-phase process, where the second phase is performed only if the first phase fails. During the first phase (called primary matching), a direct comparison is performed between the literal words used in the encodings of the option and the task. For example, if an option was "work with/schedule," and the task was "retrieve/schedule," there would be a successful primary match between the objects (schedule), but not between the actions (work versus retrieve). If primary matching fails, then secondary matching is attempted, during which a semantic net is utilized to disambiguate option words that did not have successful primary matches. This semantic net is formed of LTM elements that relate less specific words found in top-level options to more concrete words found in task descriptions. For example, if the option object was "academic information," and the task object was "schedule," then primary matching would fail. However, if the semantic net contained a link relating "academic information" to the term "schedule," then secondary matching would succeed. The use of the semantic net in PBI USER is analogous to the semantic relatedness measure used in Pierce et al. [1992]: in both cases option choice is based on the degree of semantic similarity between an option and the current task. An additional constraint that is not addressed by Pierce et al. [1992] is that in PBI USER there must exist adequate WM capacity to allow the processing of the semantic information.

If the semantic net does *not* contain the necessary knowledge to fully disambiguate the option labels at a given level of the menu, then PBI USER chooses that option with the highest degree of partial match, thus increasing the likelihood of a navigational error. If the semantic net *does* contain the necessary knowledge, then this information must be retrieved from LTM, creating additional demands for processing and storage in WM, reflecting the additional effort required to perform the disambiguation. Thus, the differences between primary and secondary matching allow us to model those differences in human interactions with the PBI-DEEP and PBI-BROAD menu structures discussed previously. The options from the upper-level menus of PBI-DEEP are less closely related to task descriptions than those of PBI-BROAD, and so will tend to require more secondary matching. If adequate LTM knowledge is found in the semantic net, then this secondary matching will result in increased demands for processing capacity. We implement the difference in strategic choice between PBI-DEEP subjects and PBI-BROAD subjects by lowering PBI USER's activation thresholds for PBI-DEEP. For example, PBI USER has a lower accept threshold for PBI-DEEP than for PBI-BROAD, resulting in a higher probability in PBI-DEEP that an option is selected before all secondary matching is completed. Therefore, PBI-DEEP may theoretically require higher processing than PBI-BROAD, but the actual processing effort exerted is less than expected. It is through the manipulation of the accept and reject thresholds that PBI USER is able to simulate different strategies for option choice and explain the pattern of results for H1, H4, H5, and H6.

8. IMPLICATIONS FOR HUMAN-COMPUTER INTERACTION

We present two main implications for the field of human-computer interaction. First, the phrase "information overload" is often discussed as a major problem in HCI, but it is rarely clear what is being overloaded. In this research we have provided a clear definition of one type of information overload (WM overload), the types of errors that may result, and a computational model of how WM capacity may be exceeded. In addition, by characterizing demands for WM resources in terms of both storage and processing (as opposed to storage only), we have gained insights into the impact of different task characteristics on WM demands. We have also gained insights into the subtle adaptations that users make in their decision-making strategies in order to cope with the task environment. PBI USER models the task environment as a combination of the tasks to be performed and the features of the interface (e.g., number of options per menu). An understanding of task complexity in terms of demand for WM storage and processing has been shown to be beneficial in explaining WM failure in human-computer interaction. By continuing to develop and refine computational models such as PBI USER, we will be able to better understand the limitations of human information processing in computer-based tasks and to account for human adaptation to overcome (or lessen) these limitations.

Second, we have shown that the use of a computational cognitive model provides a systematic approach for conducting research in human-computer interaction. But the current research also shows how difficult it is to make accurate a priori predictions based purely on the model, even for a simple task like phone-based interaction. We feel that our current state of knowledge and methodologies for building computational models are rather underdeveloped. There are multiple decisions on how to build the models and no strong guidelines. For example, it is difficult to identify those task features that, in our case, contribute to WM failure. We were correct in predicting that fewer options per menu would not result in better performance, but we failed to understand other task features such as the relationship between number of tasks and menu structure. On the other hand, we feel the computational modeling approach helped us to build an integrated explanation for results that both supported and failed to support our hypotheses.

APPENDIX

OVERVIEW OF STATISTICAL MODEL

Numbering the task response categories $l = 1$ (perfect response), $l = 2$ (task failure), $l = 3$ (rehearsal error), $l = 4$ (navigational error), and $l = 5$ (both navigational and rehearsal errors), the response of subject i to tasks m ($m = 1, \ldots, 27$) can be coded as

$$y_{iml} = \begin{cases} 1, & \text{if subject } i \text{ responds } l \text{ to task } m; \\ 0, & \text{else.} \end{cases}$$

For each subject i and each task m, $y_{im1}, \ldots, y_{im5}$ consists of four 0s and one 1, coding exactly which error (if any) was made on that task by that subject. There are 27 such five-tuples for each subject, representing the 27 repeated measures on each subject.

To describe the response behavior of a group of subjects, we begin by assuming that (a) the response is stochastic, i.e., the same subject might not give the same response to the same task on two different occasions, and (b) differences in the cognitive capacities of the individual subjects will lead to differing abilities to effectively deal with the cognitive load created by various experimental conditions. In particular we model the probability

P_{iml} that subject i makes a response of type l to task m as

$$\log \frac{\mathrm{P}_{iml}}{\mathrm{P}_{im1}} = \beta_{ml} - \theta_i, \qquad (1)$$

where β_{ml} is a parameter representing the effects of experimental conditions (task factors, menu type, reading span) on task performance, and θ_i is a parameter representing the unique, individual subject effects on task performance.

The parameters β_{ml} vary according to the experimental conditions,

$$\beta_{ml} = \phi_{cl} + \tau_{tl} + \mu_{ul} + \delta_{dl} + \rho_{rl}, \qquad (2)$$

where $c = 1, 2,$ or 3 categorizes task m according to its task format, and ϕ_{cl} is a parameter representing the effect of task format c on task error l; t and τ_{tl} represent the task order condition and its effect on task error, and similarly for u and μ_{ul} (menu) and d and δ_{dl} (minimum number of options). Finally $r = 1, 2,$ or 3 represents the reading-span group, and ρ_{rl} represents the overall propensities of the subjects in the different reading-span groups to make each kind of task error (due to the differing capacities indicated by the differing reading-span score ranges).

The model proposed in (1) and (2) is a variant of the Rasch [1980] model for psychological response data, appropriate when the responses to each stimulus are polytomous and unordered. Fischer and Parzer [1991] consider a related model for ordered polytomous response data, and Pirolli and Wilson [1992] propose a similar model to analyze learning strategies. The decomposition (2) implies that the effect of each experimental factor is the same regardless of the other factors (an "additive model"). Further terms may be added to model interactions between the factors, for example, a term $(\phi\mu)_{cul}$ for an interaction in the effect of task format and menu type on task error.

Individual differences between subjects are only partially modeled by the parameters ρ_{rl}, which describe the overall propensities of a subject from reading-span group r to make each kind of task error, regardless of the task. The parameters θ_i represent all of the other, unmodeled, features of a subject that contribute to succeeding at the tasks in the PBI domain. In particular, WM features not captured by the reading-span test, unmodeled solution strategies, etc., are represented in the statistical model by θ_i.

We assume that the probabilities P_{iml} completely characterize the response behavior of subjects, so that if we observed task performance controlling both for experimental conditions and for levels of θ_i, all task responses would be statistically independent of one another, within and between subjects. However we can only actually control for the experimental conditions and not for θ_i, since θ_i is unknown and changes with each experimental subject. This induces statistical dependence across the 27 repeated measures (tasks) within each subject. This statistical dependence is a common problem in repeated-measures designs that can be addressed in our case by averaging over possible values of θ_i.

Let $y = (y_{11}, \ldots, y_{15}, y_{21}, \ldots y_{25}, \ldots, \ldots, y_{27,1}, \ldots, y_{27,5})$ be the 27 five-tuples describing the responses of an arbitrary subject in our experiment. The model we have described above may be written formally as

$$p(y) = \int \prod_{m=1}^{M} \frac{\Pi_{l=2}^{L} \exp[y_{ml}(\beta_{ml} - \theta)]}{1 + \Sigma_{k=2}^{L} \exp[\beta_{mk} - \theta]} f(\theta) d\theta \qquad (3)$$

with β_{ml} decomposed, as in (2), into experimental factors μ_{ul}, ϕ_{cl}, τ_{tl}, and δ_{dl} and individual differences ρ_{rl} that we can explicitly control or measure. This is similar to other discrete-choice models (e.g., McFadden [1984, p. 1411]). However, individual differences between subjects are only partially modeled by the parameters ρ_{rl} in (2), which describe the overall propensities of a subject from reading-span group r to make each kind of task error, regardless of the task. Averaging over θ allows for dependence among the repeated measures (tasks) due to unmodeled individual differences between subjects. If there were other observable covariates for the between-individuals performance variation, we could construct a more psychologically accurate model by replacing θ with terms explicitly modeling this variation.

ACKNOWLEDGMENTS

We would like to thank Marcel Just, Patricia Carpenter, and Marvin Sirbu, for their comments and suggestions, and Chris Maeda, for developing the software for the experiment. Also, the valuable feedback from Wayne Gray and the three anonymous reviewers is gratefully acknowledged.

REFERENCES

AGRESTI, A. 1990. *Categorical Data Analysis.* Wiley, New York.

ANDERSON, J. R. 1983. *The Architecture of Cognition.* Harvard University Press, Cambridge, Mass.

ANDERSON, J. R. 1990. *The Adaptive Character of Thought.* Lawrence Erlbaum, Hillsdale, N.J.

ANDERSON, J. R. AND JEFFRIES, R. 1985. Novice LISP errors: Undetected losses of information from working memory. *Human-Comput. Inter. 1,* 2, 133–161.

ATKINSON, R. C. AND SHIFFRIN, R. M. 1968. Human memory: A proposed system and its control processes. In *The Psychology of Learning and Motivation.* Vol. 2. Academic Press, New York.

BADDELEY, A. D. 1986. *Working Memory.* Clarendon Press, Oxford, U.K.

BADDELEY, A. D. AND HITCH, G. J. 1974. Working memory. In *The Psychology of Learning and Motivation,* G. H. Bower, Ed. Vol. 8. Academic Press, New York.

BROADBENT, D. E. 1958. *Perception and Communication.* Pergamon Press, London, U.K.

CARD, S. K. 1988. Models of working memory. Working paper for National Research Council (Committee of Human Factors), Study on Human Performance Models for Computer-Assisted Engineering, National Research Council, Washington, D.C.

CARLIN, B. P., KASS, R. E., LERCH, F. J., AND HUGUENARD, B. R. 1992. Predicting working memory failure: A subjective Bayesian approach to model selection. *J. Am. Stat. Assoc. 87,* 319–327.

CARPENTER, P. A. AND JUST, M. A. 1989. The role of working memory in language comprehension. In *Complex Information Processing,* D. Klahr and K. Kotovsky, Eds. Lawrence Erlbaum, Hillsdale, N.J., 31–68.

COLLINS, A. M. AND LOFTUS, E. F. 1975. A spreading-activation theory of semantic processing. *Psych. Rev. 82,* 407–428.

COLLINS, A. M. AND QUILLIAN, M. R. 1969. Retrieval time from semantic memory. *J. Verbal Learn. Verbal Behav. 8,* 240–247.

DANEMAN, M. AND CARPENTER, P. A. 1980. Individual differences in working memory and reading. *J. Verbal Learn. Verbal Behav. 19,* 450–466.

DUMAIS, S. T. AND LANDAUER, T. K. 1984. Describing categories of objects for menu retrieval systems. *Behav. Res. Methods Instrum. Comput. 16,* 2, 242–248.

ENGELBECK, G. AND ROBERTS, T. 1989. The effects of several voice-menu characteristics on menu selection performance. Tech. Rep. ST0401, US West Advanced Technologies, Englewood, Colo.

FISCHER, G. H. AND PARZER, P. 1991. An extension of the rating scale model with an application to the measurement of change. *Psychometrika 56,* 637–651.

GLAS, C. A. W. AND VERHELST, N. D. 1989. Extensions of the partial credit model. *Psychometrika 54,* 635–659.

GOULD, J. D. AND BOIES, S. L. 1984. Human factors challenges in creating a principal support office system—The speech filing system approach. *ACM Trans. Off. Inf. Syst. 1,* 4, 273–298.

HABERMAN, S. J. 1977. Log-linear models and frequency tables with small expected cell counts. *Ann. Stat. 5,* 1148–1169.

HUGUENARD, B. R. 1993. Working memory failure in human-computer interaction: Modeling and testing simultaneous demands for information storage and processing. Ph.D. dissertation, Graduate School of Industrial Administration, Carnegie Mellon Univ., Pittsburgh, Pa.

JUST, M. A. AND CARPENTER, P. A. 1985. Cognitive coordinate systems: Accounts of mental rotation and individual differences in spatial ability. *Psych. Rev. 92,* 2, 137–171.

JUST, M. A. AND CARPENTER, P. A. 1992. A capacity theory of comprehension: Individual differences in working memory. *Psych. Rev. 99,* 1, 122–149.

KESSLER, C. M. AND ANDERSON, J. R. 1986. Learning flow of control: Recursive and iterative procedures. *Human-Comput. Inter. 2,* 135–166.

KIGER, J. L. 1984. The depth/breadth trade-off in the design of menu-driven user interfaces. *Int. J. Man-Mach. Stud. 20,* 201–213.

KOEHLER, K. J. AND LARNTZ, K. 1980. An empirical investigation of goodness-of-fit statistics for sparse multinomials. *J. Am. Stat. Assoc. 75,* 336–344.

LERCH, F. J. 1994. Error behavior in model building. Working paper, Graduate School of Industrial Administration, Carnegie Mellon Univ., Pittsburgh, Pa.

LERCH, F. J., MANTEI, M. M., AND OLSON, J. R. 1989. Translating ideas into action: Cognitive analysis of errors in spreadsheet formulas. In *Proceedings of the CHI '89 Conference on Human Factors in Computing Systems.* ACM, New York, 121–126.

LONGFORD, N. T. 1995. *Models for Uncertainty in Educational Testing.* Springer-Verlag, New York.

MACGREGOR, J., LEE, E., AND LAM, N. 1986. Optimizing the structure of database menu indexes: A decision model of menu search. *Human Factors 28,* 387–399.

MCFADDEN, D. 1984. Econometric analysis of qualitative response models. In *Handbook of Econometrics,* Z. Griliches and M. D. Intriligator, Eds. Vol. 2. Elsevier Science Publishers, New York, 1395–1457.

MILLER, G. A. 1956. The magical number seven plus or minus two: Some limits on our capacity for processing information. *Psych. Rev. 63,* 81–97.

MORRISON, D. F. 1990. *Multivariate Statistical Models.* McGraw Hill, New York.

PARKINSON, S. R., HILL, M. D., SISSON, N., AND VIERA, C. 1988. Effects of breadth, depth and number of responses on computer menu search performance. *Int. J. Man-Machine Stud. 28,* 683–692.

PATZ, R. J., JUNKER, B. W., LERCH, F. J., AND HUGUENARD, B. R. 1994. Analyzing small psychological experiments with item response models. Working paper, Dept. of Statistics, Carnegie Mellon Univ., Pittsburgh, Pa.

PAYNE, J. W., BETTMAN, J. R., AND JOHNSON, E. J. 1988. Adaptive strategy selection in decision making. *J. Exp. Psych. 14,* 534–552.

PAYNE, J. W., BETTMAN, J. R., AND JOHNSON, E. J. 1990. The adaptive decision maker: Effort and accuracy in choice. In *Insights in Decision Making,* R. M. Hogarth, Ed. University of Chicago Press, Chicago, Ill., 129–153.

PAYNE, J. W., BETTMAN, J. R., AND JOHNSON, E. J. 1993. *The Adaptive Decision Maker.* Cambridge University Press, New York.

PELTON, G. E. 1989. Designing the telephone interface for voice processing applications. *Speech Tech. 5,* 1 (Oct./Nov.), 18–21.

PETERSON, L. R. AND PETERSON, M. J. 1959. Short-term retention of individual verbal items. *J. Exp. Psych. 58,* 193–198.

PIERCE, B. J., PARKINSON, S. R., AND SISSON, N. 1992. Effects of semantic similarity, omission probability and number of alternatives in computer menu search. *Int. J. Man-Machine Stud. 37,* 653–677.

PIROLLI, P. AND WILSON, M. 1992. Measuring learning strategies and understanding: A research framework. In *Intelligent Tutoring Systems: 2nd International Conference, ITS '92,* C. Frasson, G. Gauthier, and G. I. McCalla, Eds. Springer-Verlag, New York, 539–558.

RASCH, G. 1980. *Probabilistic Models for Some Intelligence and Attainment Tests.* University of Chicago Press, Chicago, Ill.

SAS INSTITUTE. 1987. *SAS User's Guide: Statistics.* SAS Institute Inc., Cary, N. Carol.

SCHNEIDER, W. AND DETWEILER, M. 1988. A control architecture for working memory. In *The Psychology of Learning and Motivation: Advances in Research and Theory,* G. Bower, Ed. Academic Press, San Diego, Calif.

SMELCER, J. B. 1989. Understanding user errors in database query. Ph.D. dissertation, Univ. of Michigan, Ann Arbor, Mich.

THIBADEAU, R., JUST, M. A., AND CARPENTER, P. A. 1982. A model of the time course and content of reading. *Cogn. Sci. 6,* 157–203.

THISSEN, D. 1986. *MULTILOG User's Manual.* Scientific Software, Mooresville, Ind.

WAUGH, N. C. AND NORMAN, D. A. 1965. Primary memory. *Psych. Rev. 72,* 89–104.

Received August 1995; revised September 1996; accepted September 1996